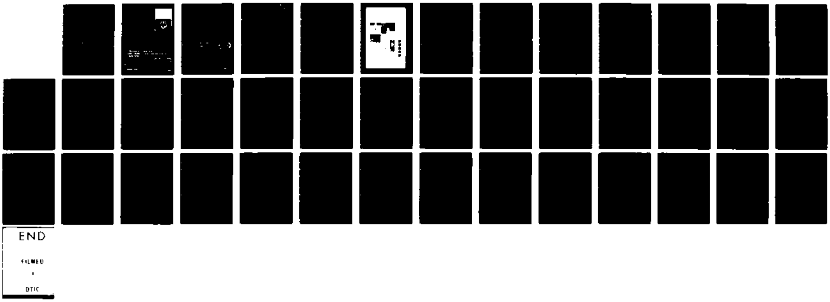
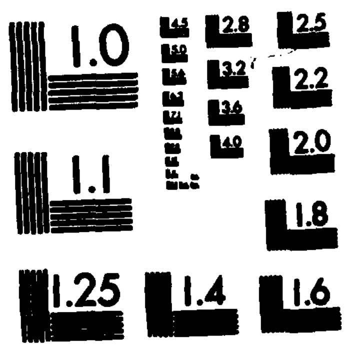
AD-A124 958 MATCHING BASED INTERACTIVE FACILITY LAYOUT(U) GEORGIA INST OF TECH ATLANTA PRODUCTION AND DISTRIBUTION RESEARCH CENTER B MONTREUIL ET AL. JUN 82 PDRC-82-02 1/1

UNCLASSIFIED N00014-80-K-0709 F/G 5/8 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(12)

Georgia Institute of Technology

AD A12[illegible]5

PRODUCTION and DISTRIBUTION RESEARCH CENTER

SCHOOL OF INDUSTRIAL
AND SYSTEMS
ENGINEERING
GEORGIA INSTITUTE OF TECHNOLOGY
A UNIT OF THE UNIVERSITY SYSTEM
OF GEORGIA
ATLANTA, GEORGIA 30332

007

PDRC REPORT SERIES 82-02
June 1982

(12)

MATCHING BASED INTERACTIVE FACILITY LAYOUT

by

Benoit Montreuil[†]
H. Donald Ratliff[††]
Marc Goetschalckx[††]

PDRC 82-02

†Industrial Engineering Section, Engineering Department,
Université du Québec à Trois-Rivières

††School of Industrial and Systems Engineering
Georgia Institute of Technology, Atlanta, Georgia 30332

This research was supported in part by the Office of Naval Research under Contract No. N00014-80-k-0709. Reproduction in whole or in part is permitted for any purpose of the U. S. Government.

The problem of "laying out" facilities is very difficult from a practical as well as a methodological point of view. As a result the layout process generally involves a "block layout" phase and a "detailed layout" phase. During the block layout phase the various elements of the facility are aggregated into areas or blocks. Each block represents a department, office, or some other major work area. An attempt is then made to optimally position these blocks within the facility. Once the block layout is determined, a detailed layout is performed. This involves specifying the exact position of equipment and work areas within each block as well as the necessary support such as electric outlets, water, etc. Except for imposing certain restrictions on the size and shape of the blocks to insure that everything will "fit," these details are essentially ignored in the block layout phase.

We develop here an interactive approach to the block layout problem. The approach has three major components: an optimization model, a colorgraphic computer interface, and a human decision maker. The subjective factors associated with evaluating designs and the combinatorial nature of the block layout problem make it impossible to model it in a form which can be optimally solved for practical problems. Hence we relax certain restrictions and optimally solve the resulting "relaxed" model. The output from the model is displayed in network form on a colorgraphics terminal. The human decision maker utilizes this information together with his knowledge of the layout problem to selectively impose additional constraints on the model or to relax previously imposed constraints. The procedure iterates between the human decision maker and the optimization model, via the colorgraphics interface, until an acceptable layout is obtained. A flow diagram illustrating the procedure is shown in Figure 1. An illustration of the colorgraphics screen layout is given in Figure 2.

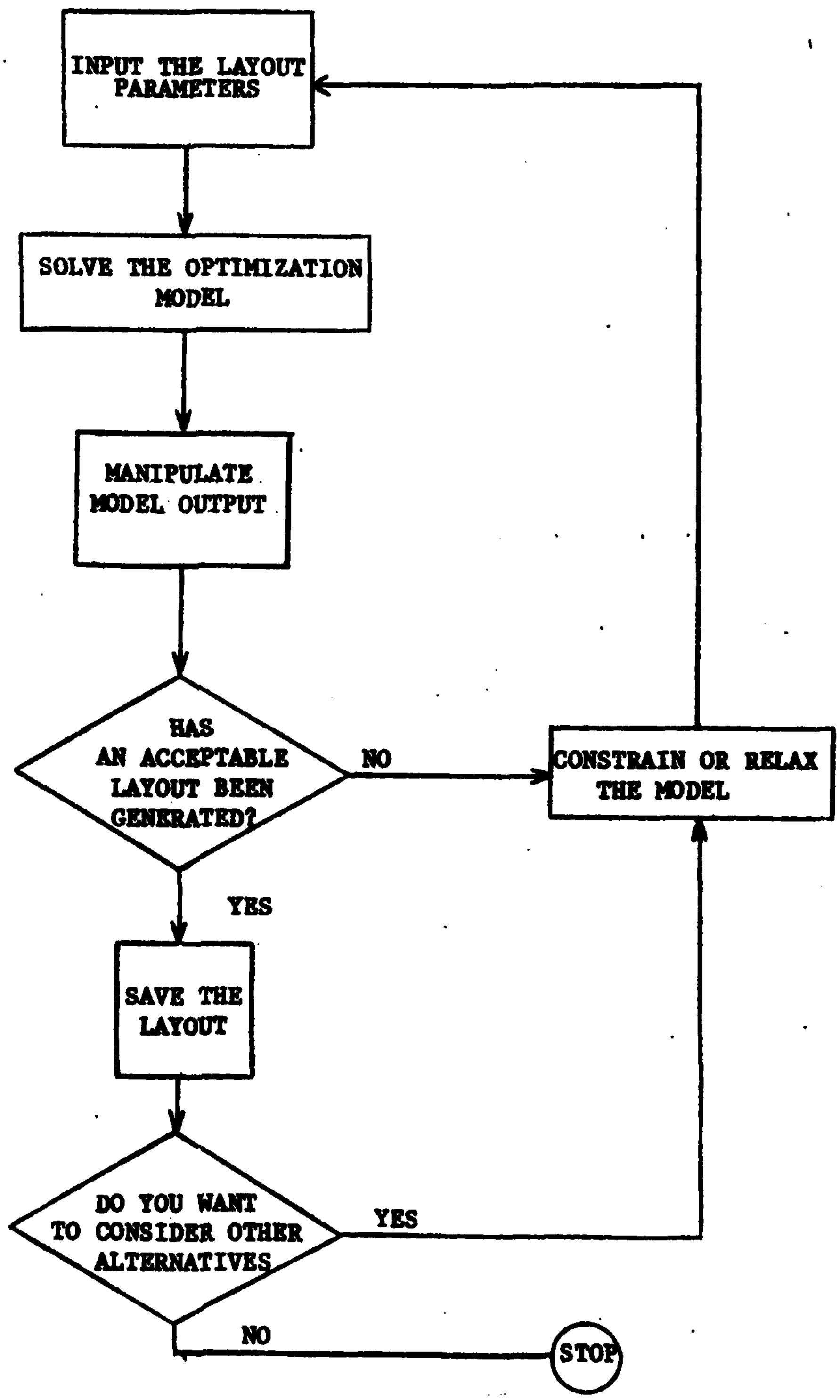


Figure 1. Flow Diagram of the Interactive Layout Process

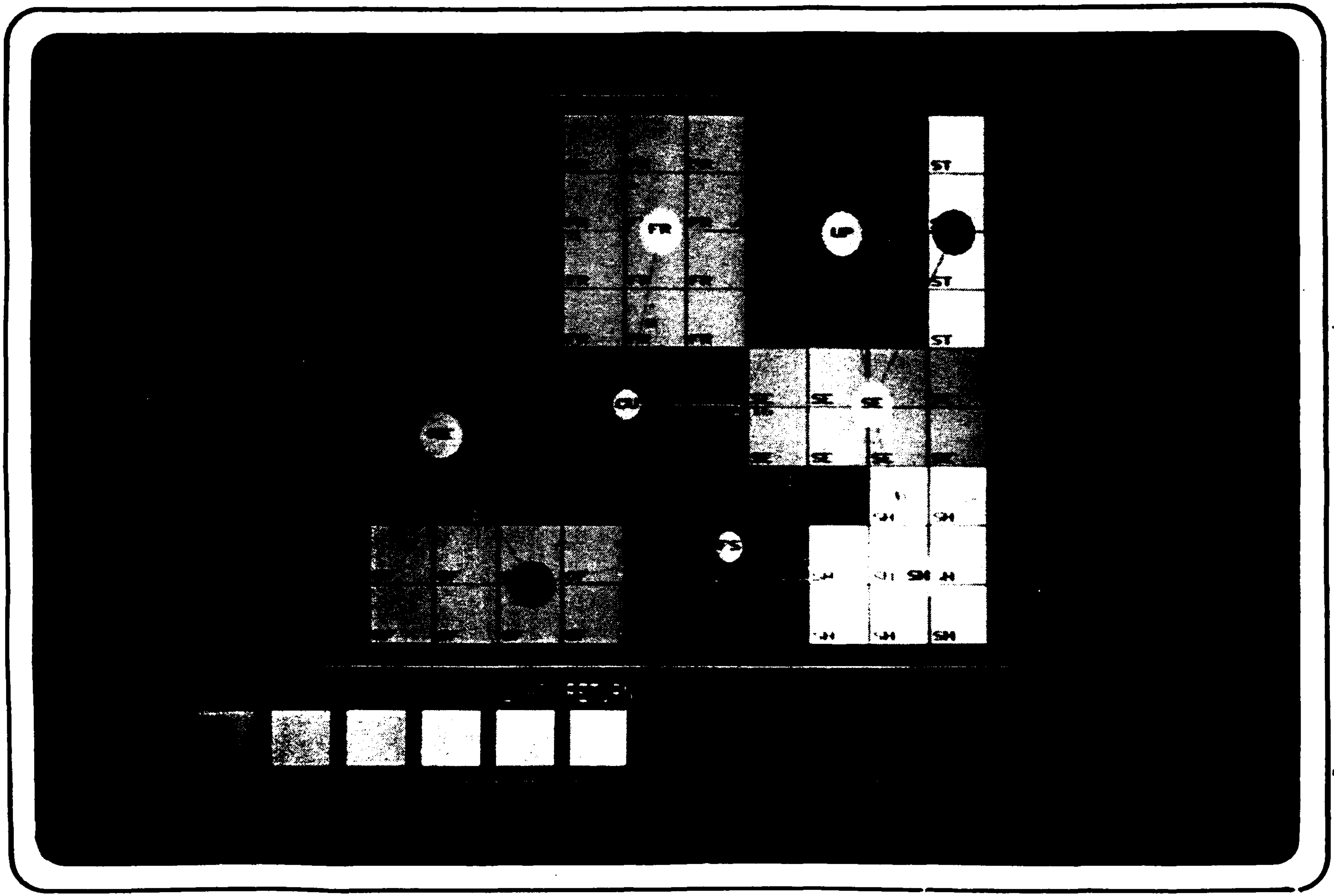


Figure 2. Illustration of the Colorgraphics Screen Layout

I. BLOCK LAYOUT PROBLEM

The block layout problem is to determine the size and shape of each block and its position within the facility. Hence, fundamental inputs to the problem are the required size and shape ranges to accommodate the equipment and workspace within the block. It should be noted that the decision as to what constitutes a block is not always easy. It is particularly difficult when the equipment and volume of work for some functions are affected by the layout (e.g. This is frequently the case when configuring storage facilities.). Even when the blocks are naturally defined by departments or work areas, there is generally some flexibility as to size and shape. Since certain block shapes are harder to utilize than others, it is often not possible to quantify this flexibility in any useful manner. However, the human decision maker can usually determine with relative ease whether or not a particular shape and size is acceptable.

As with many design problems, the quality of a layout is dependent on a large number of factors. Even for those factors which can be quantified, such as distances, it is often difficult to accurately measure them until the entire layout is completed. For example, when using modern material handling equipment, such as conveyers and wire guided vehicles which can travel only along a fixed path, the routing of material is directly dependent on the final design configuration.

In spite of this shortcoming, functions of distance are the most meaningful to work with in developing block layouts. These functions try to reflect three basic kinds of relationships: (1) between a block and the building perimeter, (2) between two blocks, and (3) between a block and some designated area in the building. Examples of blocks either desired or required to be on the perimeter include shipping and receiving, departments which have a high probability of

of expansion, and offices. Examples of desired relationships between blocks include situations where we want blocks near each other to facilitate communication and exchange of ideas and blocks between which there will be substantial material flow. Examples of desired relationships between blocks and fixed building areas include departments which because of climate requirements or weight restrictions are required to be in certain areas of the building and departments which require special equipment such as an overhead crane and must be located adjacent to it.

Estimating the form of these distance functions is easiest when considering movement of material. Even then it is a nontrivial undertaking. Generally, the further material is to be moved, the more costly it is to move it. However, the manner in which it is to be moved can depend heavily on the distance it is to be moved. In particular, movement between two adjacent departments can often be done much more cheaply and with different equipment than movement between nonadjacent departments even if their separation is small. Also the cost of movement between two departments, particularly if they are not adjacent, is often dependent on other movement within the facility. Hence, the distance functions may be at best crude approximations of the desirability of having two areas close to each other.

II. LAYOUT PROGRAM

Since the middle 1960's a number of efforts have been made to develop computer aided layout programs. Five better known procedures are detailed in Tompkins and Moore [4]. These procedures have been categorized as either "constructive" or "improvement" procedures. The construction procedures, (CORELAP, ALDEP, and PLANET) attempt to build a layout by a one-at-a-time insertion of blocks into the layout. The first block to be entered is either selected at random or selected based on some heuristic measure of importance. Subsequent blocks are selected and positioned based on heuristic scoring schemes which relate them to blocks already placed in the layout. While these scoring schemes seem reasonably logical, they have no overall model or methodology as a basis. The improvement procedures (CRAFT and COFAD) start with a block layout and attempt to improve it by interchanging blocks. These interchanges are generally limited to interchanges between blocks of the same size or blocks which are adjacent.

All of these programs have some serious limitations. They either ignore the desired shapes of the blocks or very crudely consider them. Other than possibly fixing a block at some prescribed location, they do not consider relationships between blocks and the building perimeter or between blocks and fixed areas of the building. They have very little methodological base and they allow little flexibility for the designer to influence the layout using his insight and ingenuity.

Several attemps have been made to develop block layout methods based on optimization models. These methods are surveyed in a recent paper by Foulds [2]. In general they are based on either the quadratic assignment problem or the problem of finding a maximum weighted planar graph. Neither of these

problems is tractable for problems of practical size. These models are in general relaxations of the actual layout problem, and the heuristics for solving them do not seem particularly helpful in providing the human decision maker with insight to develop feasible layouts.

The methodology developed here is an attempt to overcome some of the difficulties associated with current methodology. In particular, the emphasis is on providing the human with information which will aid him in constructing good layouts rather than having the computer construct a layout.

III. The Matching Model

As indicated earlier, there is no optimization model which captures all of the characteristics of the layout problem and remains tractible. Hence, in order to solve the optimization model we must ignore to a degree (i.e., relax) at least some of the constraints on the problem. If the solution to the relaxed problem is not acceptable in terms of the layout, then the human decision makers must somehow restrict or constrain the model to force it toward an acceptable solution. Also, the constraints imposed, when added to the model, must result in a new model which remains tractable. Otherwise, the optimization model is of little value. An optimization model which we have found particularly attractive for this purpose is the b-matching model of Edmonds [1]. We will refer to the model as simply the matching model.

Graphical Interpretation

The matching model is easiest to understand when interpreted on a graph. Consider a graph having vertices $v_1, v_2, \ldots, v_n$ and edges e_{ij} connecting each vertex pair v_i and v_j. Each edge e_{ij} has an integer lower bound l_{ij} and an integer upper bound u_{ij}. The lower and upper bounds restrict the number of times the edge can be used. If edge e_{ij} is used k times, then vertices v_i and v_j are said to be "matched" k times. A weight of w_{ij} is associated with each matching of v_i and v_j. For each vertex v_{ij} there is an integer b_i, which specifies the total number of times v_i can be matched with all other vertices. We call this the "degree" constraint. The objective is then to find a matching which satisfies the constraints and maximizes the sum of the weights.

As an illustration, consider the graph in Figure 3. For the upper and lower bounds and weights indicated, the optimum solution is to match v_1 and v_2 one time, v_1 and v_4 one time, v_2 and v_3 three times, v_2 and v_4 zero times, and v_3 and v_4

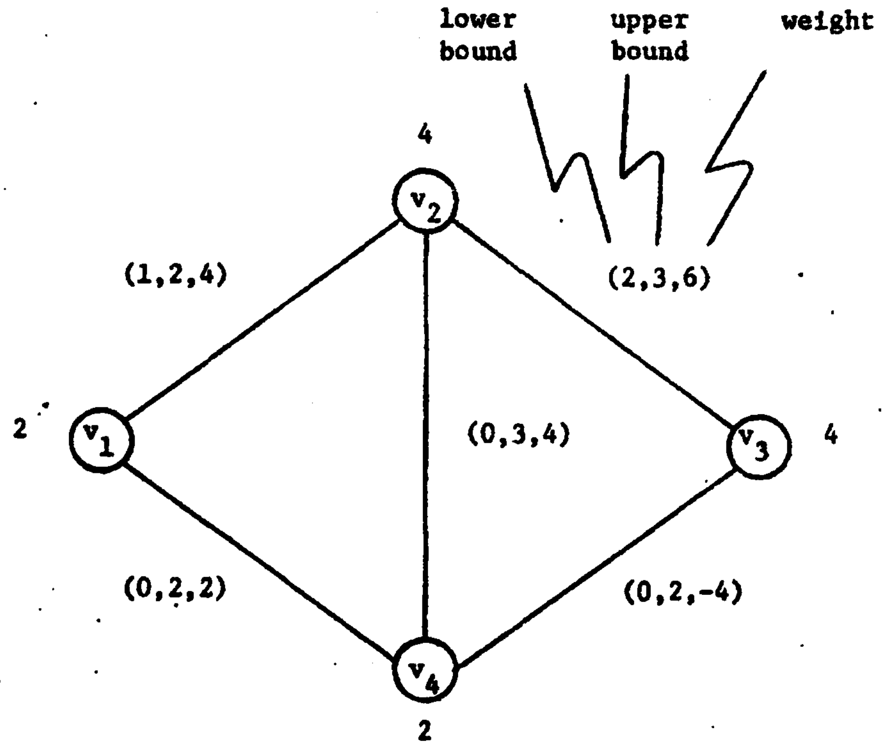


Figure 3. Example Matching Problem

one time. The total weight is 4 + 2 + 18 - 4 = 20.

Algebraically, this problem can be stated as

$$\begin{aligned}
\text{Maximize} \quad & 4x_{12} + 2x_{14} + 6x_{23} + 4x_{24} - 4x_{34} \\
\text{subject to} \quad & x_{12} + x_{14} = 2 \\
& x_{12} + x_{23} + x_{24} = 4 \\
& x_{23} + x_{34} = 4 \\
& x_{14} \quad x_{34} = 2
\end{aligned}$$

$$1 \le x_{12} \le 2 \quad 0 \le x_{14} \le 2 \quad 2 \le x_{23} \le 5 \quad 0 \le x_{24} \le 3 \quad 0 \le x_{34} \le 2$$

$$x_{12},\ x_{14},\ x_{23},\ x_{24},\ \text{and } x_{34} \text{ integer}$$

Edmonds [1] has developed a very efficient algorithm for solving problems of this form. For the special case where b_i, ℓ_i and u_i are all even for $i = 1,2,\ldots,n$, we can ignore the integer constraints and simply solving the resulting linear program. The simplex method will always give integer values for the x_{ij} and the sensitivity information normally associated with linear programs is valid.

Modeling Block Layout

To see how the matching model can be used to aid in solving the block layout problem, consider first the case where all departments are square and have the same dimensions. Also, assume that the building in which the departments are to be located has a fixed rectangular shape. A possible layout for a four department example is shown in Figure 4.

The perimeter of each department can be thought of as four equal length segments, one corresponding to each wall. The perimeter of the building in Figure 2 can be thought of as eight segments, each of the same length as the side of a department. For any layout of the four departments within the building, each of the four perimeter segments corresponding to a department must be adjacent to either a perimeter segment for another department or a perimeter segment of the building. Hence, we can think of the layout in terms of "matching" the perimeter segments of a department with either perimeter segments of other departments or perimeter segments of the building. It is this concept which gives rise to the matching model as a tool for block layout.

We can develop a matching model by constructing a vertex corresponding to the perimeter of each department and a vertex corresponding to the perimeter of the building. The degree constraint on each vertex represents the number of perimeter segments for the corresponding block. The graph for the example in Figure 4 is shown in Figure 5. The E vertex represents the building perimeter and has a degree constraint of eight. Each of the vertices corresponding to departments has a degree constraint of four. Edges are constructed between each pair of vertices. The lower and upper bounds represent the minimum and maximum number of perimeter segments the corresponding blocks can have adjacent. For example departments A and B can have a minimum of zero and a maximum of one perimeter segments adjacent. The edge weights are a measure of the desirability of having two blocks adjacent. A negative weight means that it is undesirable to have the two blocks adjacent. The optimum solution to the matching model is given in Figure 6 and the corresponding layout in Figure 7.

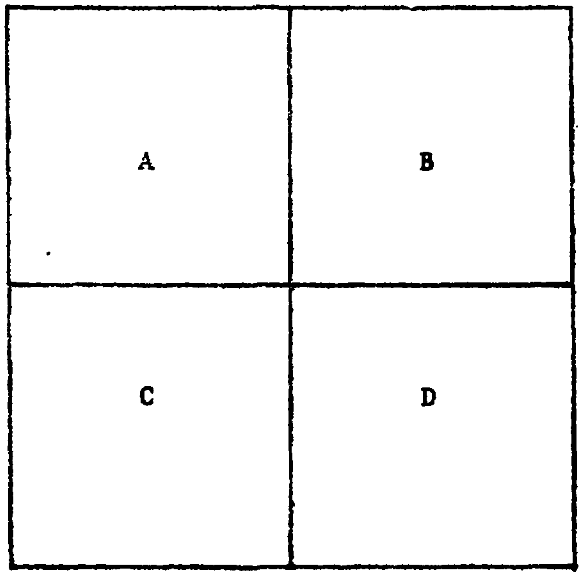


Figure 4. A Block Layout Example Where All Departments Are Squares of the Same Dimensions.

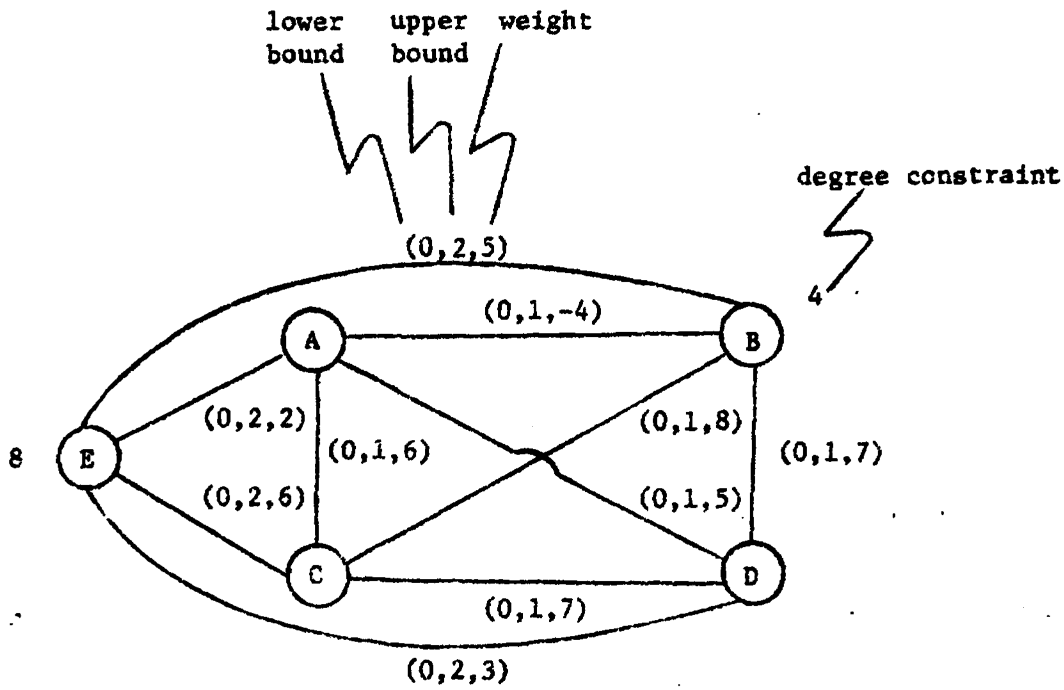


Figure 5. Matching Model Corresponding to the Example in Figure 3.

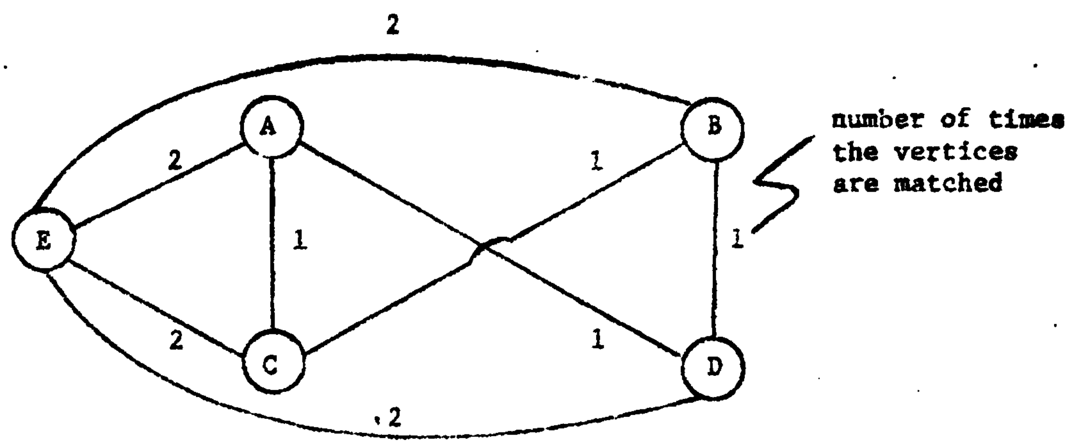


Figure 6. Optimum Matching Solution for the Problem in Figure 4.

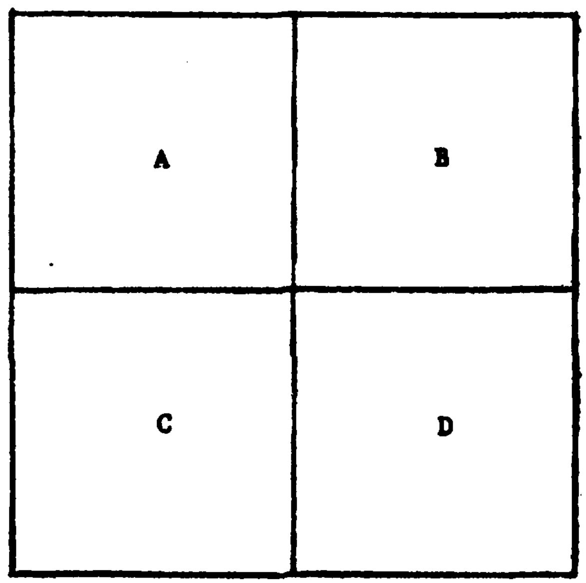


Figure 7. Layout Corresponding to the Matching Solution in Figure 6.

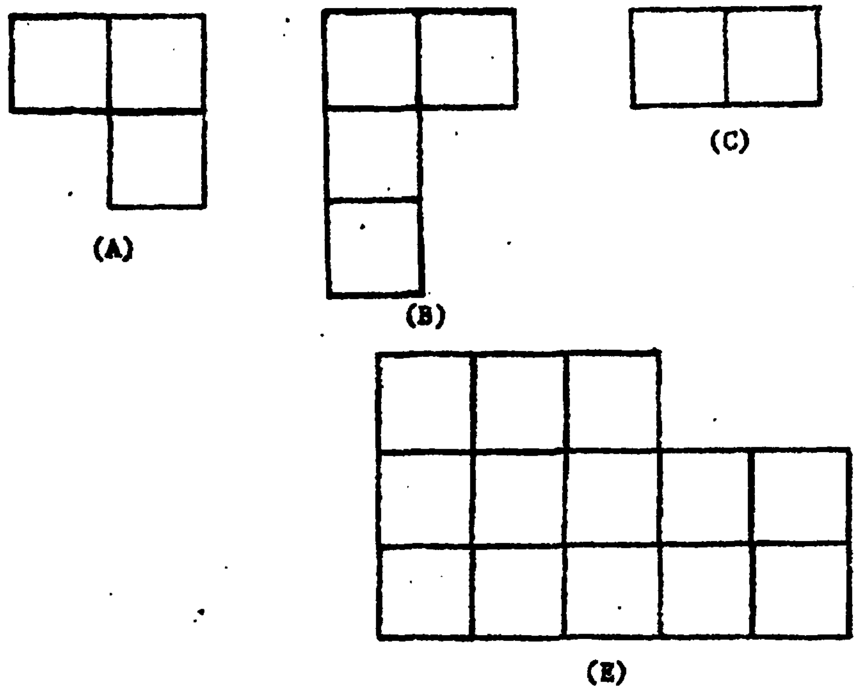


Figure 8. A Block Layout Example Where Blocks are not Necessarily Square.

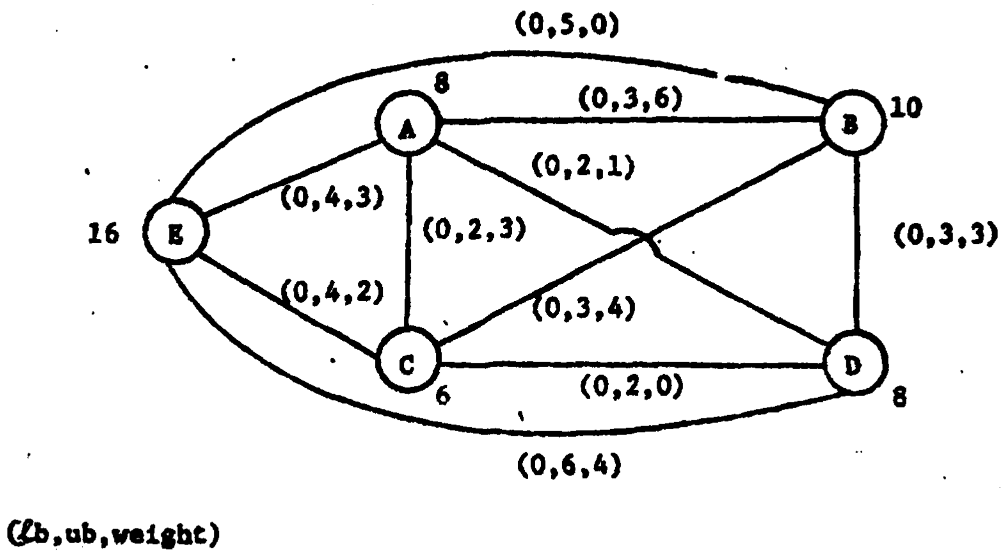


Figure 9. Matching Model for the Example in Figure 8.

In general the matching model is a relaxation of the layout problem. Hence, it may not be possible to construct a feasible layout corresponding to a solution to the matching model. Before addressing the consequence of the relaxation, we will first consider how to model different block shapes and how to specify edges weights. Consider the blocks in Figure 8. Blocks A, B, C, and D are departments and block E is the building. Each block is constructed of "unit squares." The edges of each unit square which lies on the perimeter of the block is considered as one perimeter segment. Block A has 8 perimeter segments. Hence, the corresponding node in Figure 9 has a degree constraint of 8. Degree constraints for any block made up of unit squares can be similarly determined.

If the blocks do not completely fill the building, we can still use the matching model if we define "dummy" blocks, each consisting of one unit square, to fill the excess area. The weights between these dummy blocks and all other blocks is zero.

Furthermore, we can add flexibility in department and building shapes, if desired, by use of "slack" edges, which have the same effect as slack variables in linear programming. For example, suppose the required building area is 100 and we want it to have a shape with a maximum length/width ratio of 4. If the building is square (10x10) the corresponding node degree would be 40. The other extreme is a 5x20 rectangular building with a node degree of 50. By fixing the building node degree to 50 and adding a slack edge to this node with capacity 10 we can allow a range of shapes between these two extremes. This can also be done to add flexibility to any block shape. Weights can be associated with the use of a slack edge to impose a marginal cost for each extra unit of perimeter added to the minimal required perimeter.

The upper bounds on edges correspond to the maximum number of times that two blocks can be matched. For example, in Figure 6 blocks B and C can be matched a maximum of 3 times (i.e., rotate C ninety degrees and make it adjacent to the lower right portion of B). Hence, the upper bound on the edge from B to C in Figure 7 is 3. If we wish to exclude the possibility of B and C being oriented in this fashion, we can make the upper bound less than 3. All upper bounds in Figure 7 are at their maximum.

Lower bounds are zero unless we wish to force two blocks to be adjacent. By specifying the lower bounds we can force the model to match as many segments between two blocks as desired.

Note in Figure 9 that all degree constraint are even. It is easy to show that this will always be the case for blocks made up of unit squares. By redefining a unit square to be one-fourth of the area of the original unit square, we can make all of the l_j and u_j even as well. This will cause each of the original b_i, l_j, and u_j to be multiplied by two. Hence, we can solve the matching model corresponding to a layout problem using the simplex method and be assured of integer answers.

The matching model allows some flexibility in specifying the edge weights. For a given edge it only requires that the weight function be nonincreasing for each additional time two vertices are matched. For example, in Figure 9 the weight of 6 on edge A-B indicates that a weight of 6 is obtained for each segment that A and B have in common. The matching model would also allow us to use a weight of 6 for the first segment that A and B have in common, 4 for the second segment and zero for the third.

For the problem considered to date, we have selected weights corresponding to the amount and cost of material flow between each pair of blocks. If one

perimeter segment in common is sufficient to allow the material flow between the two blocks, then the total weight can be given to the first segment matched and zero to the others. If more than one segment in common is required for material flow, then the weights can be allocated among the segments as long as they are nonincreasing.

It is important to remember that the output of the matching model will not be a layout but rather will be information to help the human decision maker construct a layout. Hence, the main function of the weights is to influence the matching solution rather than to provide a score for the layout. This will be discussed in more detail in a later section.

Relaxation Implications

The matching model is a relaxation of the block layout problem. This means that for every feasible block layout there is a corresponding feasible solution to the matching model. However, some feasible solutions to the matching model may not have corresponding feasible block layouts. Hence, the matching model does not capture all of the characteristics associated with the shapes of the blocks and how they fit together. Since the human decision maker wishes to use the matching model output to aid in construction of the layout, it is necessary to understand what characteristic have not been captured. They can be considered in three major categories which we will call planarity, grid infeasibility, and adjacency dependence.

Suppose that we take a solution to the matching model and draw the resulting graph, excluding edges which are not in the matching. A necessary condition for there to be a block layout corresponding to this solution is that we be able to draw the graph with the building perimeter vertex on the outside and have no two edges cross (i.e., the graph must be planar). We call this the

planarity restriction.

As an illustration of the grid restriction, consider three blocks each consisting of a single unit square. Because of our restriction that these blocks be laid out in a grid, it is not possible to have all three mutually adjacent.

Adjacency dependence occurs when the adjacency of two blocks is dependent on whether or not they are adjacent to certain other blocks. For example, in a rectangular building a block can be placed in a corner position only if at most three other blocks are in corner positions.

Since the original matching model cannot explicitly include these constraints the solution to the matching model may not lead directly to a feasible layout. We attempt to overcome this difficulty by having the human decision maker look for violation of these constraints by the matching solution and then impose constraints to force the matching solution toward a feasible layout. For example, if the solution to the matching model of a rectangular building case has five blocks indicated in building corners, we could pick one block and restrict it to not be in a corner. The change in the value of the matching solution uder the new constraint gives a bound on how much effect this constraint imposed on the optimum layout. The various constraints and how they are implemented in the MATCH system will be discussed in the next section.

Imposed Constraints

All of the constraints which we allow to be imposed on the matching model amount to changes in the upper and lower bounds on edges. For example, if we have a block consisting of a single unit square and we decide not to allow it in a corner, we simply make the upper bound one on the edge between its node

and the building perimeter node. If we want to force two edges not to cross, we can restrict one of the edges to an upper bound of zero, hence eliminating it from the graph. The various restricting functions we allow in the system are summarized below.

We can restrict a block as to whether or not it is a corner block, an outside block but not a corner, or an interior block. We can also restrict a block to a particular location within the building. Finally we can restrict the upper and lower bounds on the adjacency between any pair of blocks. All of the restrictions are handled by changing parameters in the matching model.

IV. MATCH SYSTEM

The MATCH system is implemented using a Chromatics colorgraphics terminal together with a CYBER mainframe computer. Data input and manual manipulation of the blocks of the layout is handled on the Chromatics. The matching model is implemented on the CYBER. The CYBER is accessed automatically each time we wish to solve a new matching problem.

The system is run utilizing a light pen. The functions are accessed by touching the associated menu box with the light pen. The blocks in the layout as well as the matching graph are manipulated using the light pen. For example, to shift a vertex from one position to another on the colorgraphics screen, you touch the function SHIFT on the screen, then touch the vertex, and then touch the new location for the vertex. The vertex is automatically shifted and the graph redrawn. Manipulation of the blocks is handled in a similar fashion. A sample of a screen layout is shown in Figure 2. Colors are used to convey information about the matching solution (e.g., red vertices indicate that the corresponding block should be a corner block) and to distinguish one block from another.

A listing of the various functions which have been incorporated into the MATCH system are given in Table 1 along with a brief description of each. The menu structure is given in Table 2. For example, if you touch the INPUT function in the main menu, the INPUT menu is drawn for you from that menu you can access any of the functions DATA, MODIFY, SAVE, LOAD, TRANS, as RETURN simply by touching the appropriate function with the light pen.

Table 1. Functions Available Within the MATCH System

0	MAIN	main menu, automatically displayed
1	INPUT	input menu, original data input phase
2	GRAPH	graph menu, relaxation and constraint phase
3	LAYOUT	layout menu, block layout phase
4	CRAFT	execution of improving CRAFT algorithm
5	EXIT	termination of the program
6	DATA	input of original dimensions, names and cost of departments
7	MODIFY	modification menu of input data
8	SAVE	save the current problem on disk
9	LOAD	reload a saved problem from disk
10	TRANS	transmit input or graph data for execution of matching algorithm
11	RETURN	return to higher menu
12	NAME	change name and description of a department
13	DIMEN	change dimensions of a department
14	RELAT	change relationship between 2 departments
15	LIST	list in a round robin fashion a) input data b) result from matching c) added constraints
16	DRAW	draw menu to draw graph based on result from matching
17	MODIFY	modify menu to change graph and/or add constraints
18	ADD	draw one additonal user selected department
19	SHIFT	shift department vertex in graph
20	AUTO	draw automatically all not-yet-drawn vertices
21	REDRAW	redraw current displayed screen (graph or layout)
22	NODE	change or specify adjacency between department and the outside
23	EDGE	edge modification menu to change or specify adjacency between 2 departments
24	SHOW-∅	draw all arcs fixed to a zero-adjacency
25	UP BND	specify upper bound on adjacency between 2 departments
26	LO BND	specify lower bound on adjacency between 2 departments
27	FIX	fix the adjacency between 2 departments
28	INFO	display arc and adjacency-parameters between 2 departments
29	BUILD	build menu to construct a block layout based on graph data
30	MANUAL	build menu to construct a block layout without graph data
31	SCORE	score a complete layout with different measures
32	NODE	node menu to change or act upon a department in the graph
33	BLOCK	block menu to act upon a department in the block layout
34	INFO	display information about a department and its adjacencies
35	JOIN	permanently link a node representation to a block representation of a department
36	GLOBAL	place a complete department in the block layout
37	SINGLE	place a single (1x1) grid block of a department in the block layout
38	DELETE	delete a single (1x1) grid block of a department from the layout
39	SHIFT	shift or relocate a single (1x1) grid block of a department
40	ERASE	delete a complete department from the layout
41	COLOR	change the color of a located department

Table 2. Menu Structure for the MATCH System

MAIN (0)	INPUT(1)				
	GRAPH(2)				
	LAYOUT(3)				
	CRAFT(4)				
	EXIT(5)				
INPUT(1)	DATA(6)	GRAPH(2)	DRAW(16)	LAYOUT(3)	BUILD(23)
	MODIFY(7)		MODIFY(17)		MANUAL(30)
	SAVE(8)		SAVE(8)		SCORE(31)
	LOAD(9)		LOAD(9)		SAVE(8)
	TRANS(10)		TRANS(10)		LOAD(9)
	RETURN(11)		RETURN(11)		RETURN(11)
MODIFY(7)	NAME(12)	DRAW(16)	ADD(18)	BUILD(29)	NODE(32)
	DIMEN(13)		SHIFT(19)		BLOCK(33)
	RELAT(14)		AUTO(20)		REDRAW(21)
	LIST(15)		REDRAW(21)		SCORE(31)
	RETURN(11)		LIST(15)		RETURN(11)
			RETURN(11)		
				NODE(32)	INFO(34)
		MODIFY(17)	LIST(15)		SHIFT(19)
			SHIFT(19)		JOIN(35)
			REDRAW(21)		RETURN(11)
			DIMEN(13)		
			NODE(22)	BLOCK(33)	GLOBAL(36)
			EDGE(23)		SINGLE(37)
			RETURN(11)		DELETE(38)
					SHIFT(39)
		EDGE(23)	RELAT(14)		ERASE(40)
			UP BND(25)		COLOR(41)
			LO BND(26)		RETURN(11)
			FIX(27)		
			SHOW-Ø(24)	MANUAL(30)	GLOBAL(36)
			INFO(28)		SINGLE(37)
			RETURN(11)		DELETE(35)
					SHIFT(39)
					INFO(34)
					COLOR(41)
					RETURN(11)

V. MATCH SYSTEM EXAMPLE CASE

This example case is based on a real situation encountered in a small Canadian company which makes and sells a variety of aluminium signs, coupled with low volume domestic aluminium products. This company was planning to relocate its plant in a new building. The data was gathered through a systematic analysis of all products, parts, material, and employee movements. It is used here to illustrate a typical execution of the MATCH LAYOUT system. It should be understood that there is no "best" procedure for using the system. It is specifically designed to allow the user to alter the approach depending on the particular application.

Design Information

The plant is divided in 12 departments. In Table 1 are the proposed shapes and sizes for each area in relative dimensions (i.e., all dimensions are multiples of the width of one unit square). In Table 2 we present in non-increasing order the no-zero exchanges (trips per day) between departments, as found by plant analysis. Note that an exchange with the building perimeter should be translated as an exchange with the outside environment.

Iteration 1

Once the data is entered into the MATCH system and saved, we transmit the problem data to the matching program in order to get a first solution. When the solution comes back, we command an automatic drawing of the resulting graph. We then manually manipulate this graph in an attempt to reduce the edge crossing, while generally putting the vertices in the suggested proximity to each other and to the perimeter. The result is given in Figure 10. The matching solution has a value of 319. This gives an upper bound (under the matching objective)

Table 3. Proposed Shapes and Sizes of Departments for the Layout Case

Department	X and Y dimensions
SILKSCREEN (SE)	1, 2
SECONDARY TRANSFORMATION (ST)	1, 2
OFFICES (OF)	1, 1
MAIN TRANSFORMATION (MT)	2, 2
SCOTCHLITE (SC)	1, 2
FILM PREPARATION (FP)	1, 2
WAREHOUSE (WH)	1, 2
SHIPPING (SH)	1, 1
OVEN ROOM (OV)	1, 1
SPECIAL STOCK ROOM (SS)	1, 1
PAINTING ROOM (PR)	1, 1
EMPLOYEES SERVICES (ES)	1, 1
BUILDING (BU)	4, 5

Table 4. Exchanges Between Department for the Layout Case

DEPARTMENT PAIR	EXCHANGE	DEPARTMENT PAIR	EXCHANGE
OV, PA	58	MT, ES	2
SE, OV	47	SC, SH	2
MT, OV	29	SC, ES	2
SE, FP	28	FP, ES	2
SH, OV	27	WH, SH	2
SC, OV	20	SH, SS	2
SE, SC	12	SH, ES	2
ST, MT	7	SH, BU	30
SC, PA	7	WH, BU	15
SE, MT	5	OF, BU	10
SE, ES	2	MT, BU	10
ST, WH	2	ES, BU	10
ST, ES	2	FP, BU	5
OF, SH	2	OV, BU	5
OF, SS	2	PA, BU	5
MT, WH	2		

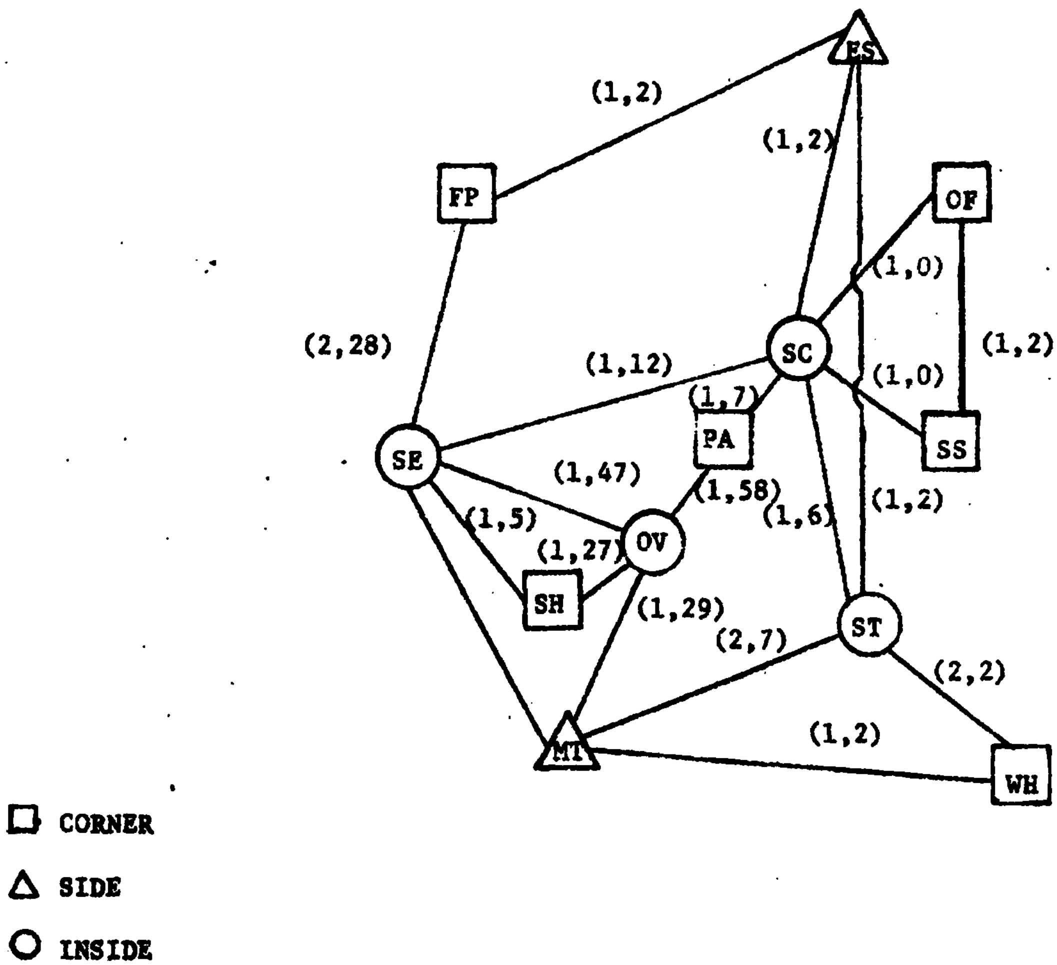


Figure 10. Solution to the First Matching Problem After Manual Manipulation

while shapes (i.e., squares, triangles, and circles) are used to indicate the proximity of the vertices to the perimeters in Figure 10, colors, (red, yellow, green) are used in the system itself.

This first graph is obviously not a feasible layout with its six corner departments in a rectangular plant. However it permits us to learn some interesting facts. The most important is the dominance of the oven room around which departments (painting room, silkscreen, main transformation and shipping) have been centralized due to their high exchanges. It seems logical at this point to restrict the oven room to be interior to the building. We note that the shipping and painting departments also have large non-zero exchange with the building perimeter. This fact has caused the model to put them in corners while being adjacent to the oven room. Since they all have 1 x 1 dimensions, this is physically impossible if the oven room is in the interior. Coupling this fact with the observation that for the shipping and painting departments, it would probably be sufficient to have only one side adjacent to the exterior, we restrict both of them not to be in corners.

At that point we could have stopped analyzing the graph and immediately asked for a revised matching result. However we decided to pursue our analysis in order to give it, if possible, a little more structure. This has been done by analyzing the trio (warehouse, main and secondary transformation) which has been joined by the matching. Secondary transformation has its only exchanges with the main transformation (7) and the warehouse (2), while the warehouse has an exchange with the main transformation (2) and the shipping (2); the main transformation and the warehouse both have strong exchanges with the exterior. This led us to try to fix their configuration (MT (2,2), ST (1,2), WH (1,2)) to be as shown in Figure 11. This implies that we fix MT and WH as

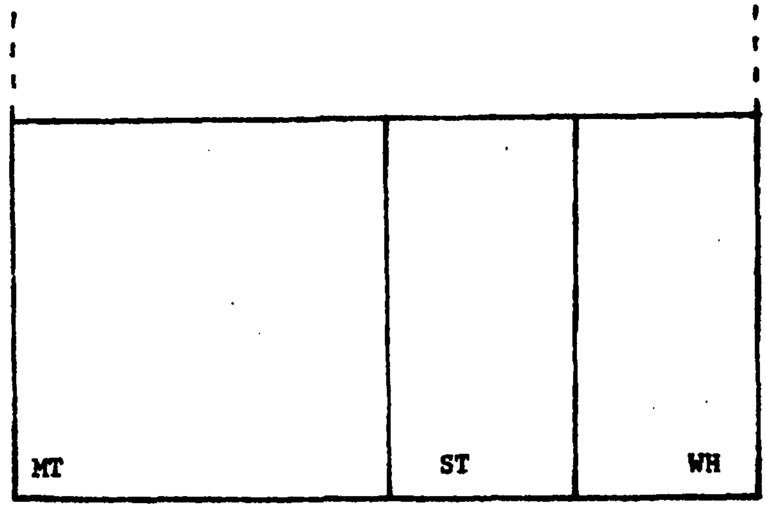


Figure 11. Desired Configuration for the MT, ST, and WH Blocks

corners, ST to have its small side on the building perimeter, MT and ST to have an adjacency of two, ST and WH, to have an adjacency of two, and MT and WH to have an adjacency of zero. Note that this is only a trial configuration which can be eliminated or modified if it appears to be too restricting later on. With these restrictions imposed we now ask for a revised matching result.

Iteration 2

The matching solution under the above restrictions is shown in Figure 12. The new solution value is 319. This means that the restrictions which we imposed have not decreased to solution value. Hence, we are inclined to retain the restrictions.

We note that there are still some problems to overcome before we have a feasible layout. This time we decide to go a little further in structuring the oven room centralization. By analyzing the graph, we arrive to the following restrictions. The adjacencies between (SH and OV), (OV and PA) and (OV and MT) are fixed to one. The trio (SH, OV, PA), all 1 x 1 departments, are laid out in a line alongside MT and ST; with the shipping department on the outside. (see Figure 13) So (SH and MT) and (PA and ST) are set to one. SH is forced to have one side on the outside while PA is forced to be in the interior. Let us note again that this is only a trial configuration which appears logical. It can be eliminated or modified later if we find it has undesirable consequences. We now ask for a revised matching solution.

Iteration 3

The new matching solution has a value of 308 and is shown in Figure 14. This graph almost gives directly a feasible layout. In fact a large part of the layout can be drawn without infeasibility as indicated in Figure 15.

The only violations are:

OF is not a corner, as prescribed

FP and ES are not adjacent

FP and SC are adjacent

SE and SC only have an adjacency of one

This layout has a score of 306. Hence, we have constructed a layout which is only about 4% less than the bound determined by the first matching solution.

Iteration 4

We save the layout above and then proceed to a careful analysis of the adjacencies and exchanges to determine whether or not we can come up with some improvement possibilities. A layout with a slightly better score is found by noting that changing from the configuration (MT, ST, WH) to (WH, MT, ST) loses nothing in the trio, and gains the adjacency between WH and SH. This leads to the layout in Figure 16.

At this point, if we are happy with one of these layouts, we can terminate the process. If we want to generate additional layouts, we can do so by starting from any of the matching solutions determined in the previous iterations. We can also fix any part of the current layout that we like and use it as a starting point. Typically, we can generate a number of different layouts with about the same scores. The decision as to which of them to select can then be based on nonquantitative factors.

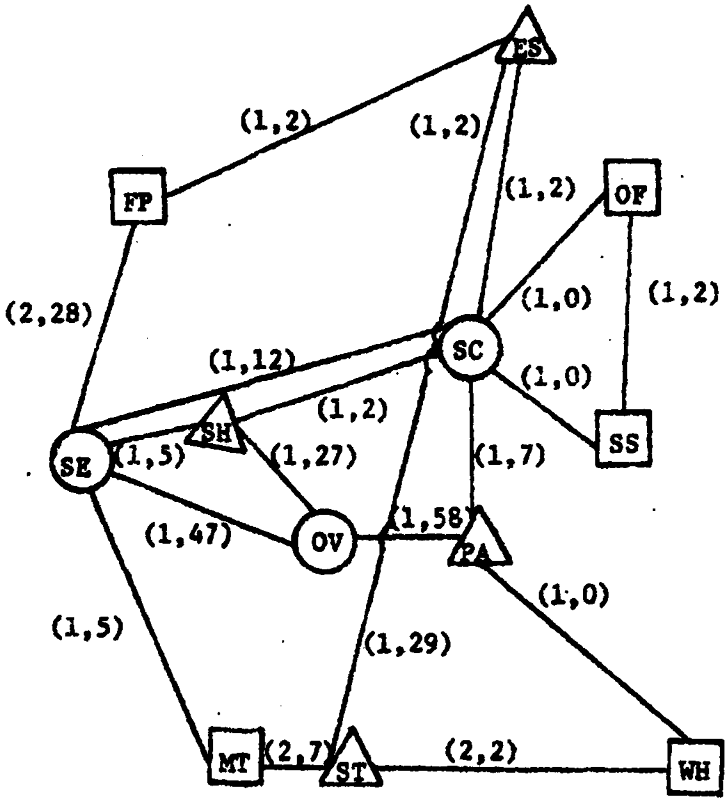


Figure 12. Solution for the Iteration 2 Matching Problem

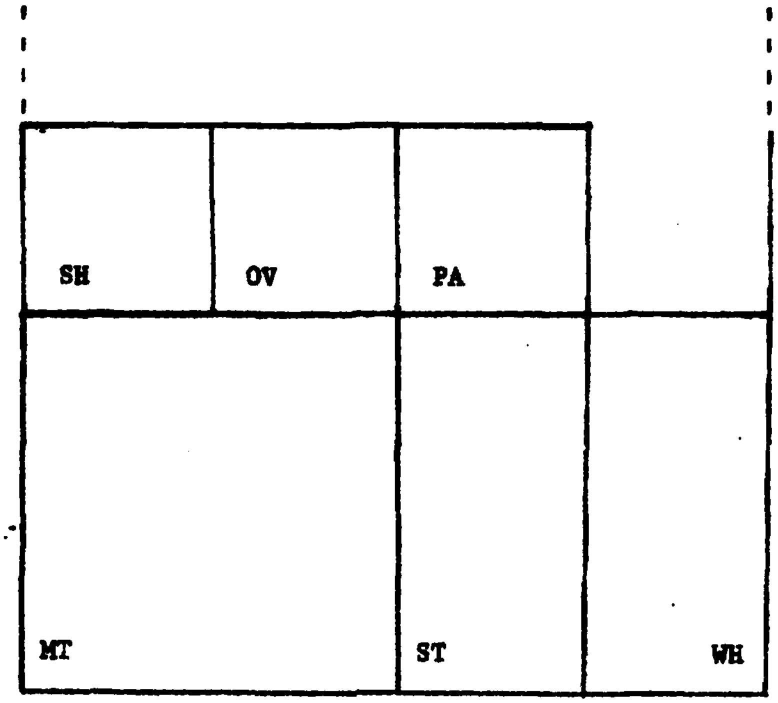


Figure 13. Desired Configuration for the SH, OV, PA, MT, ST, and WH Blocks

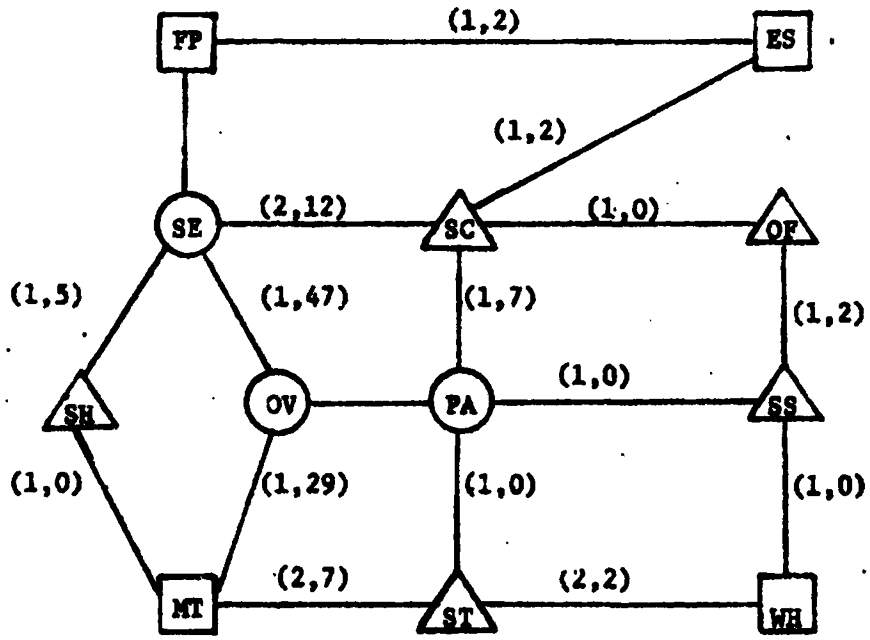


Figure 14. Solution for the Interation 3 Matching Problem

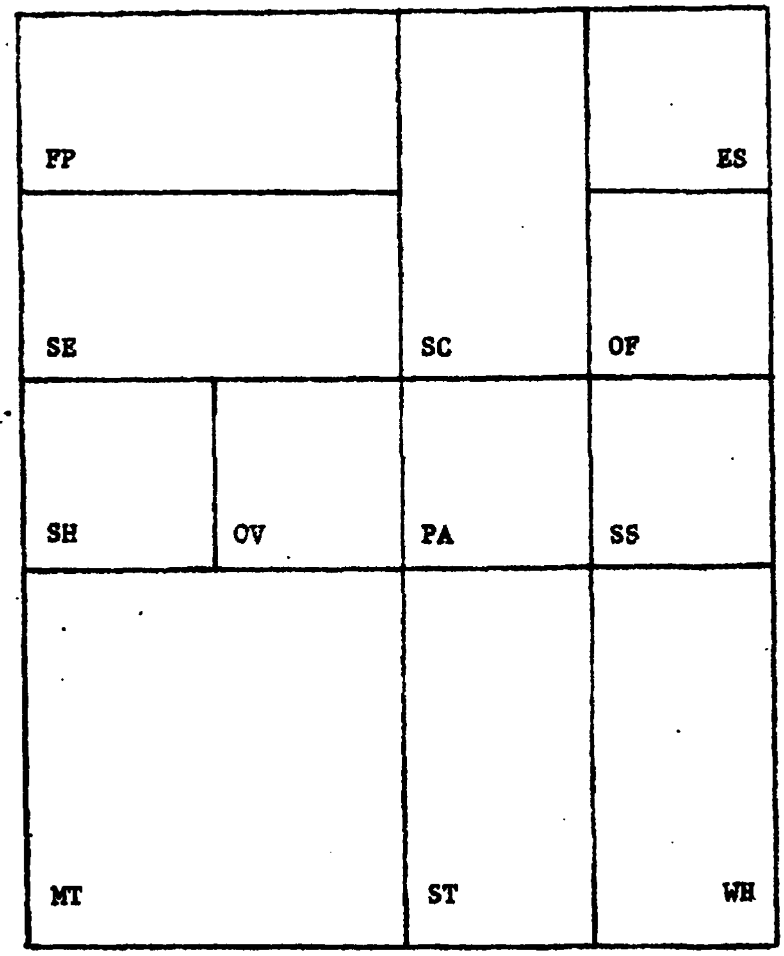


Figure 15. A Layout Close to That Indicated by Figure 14.

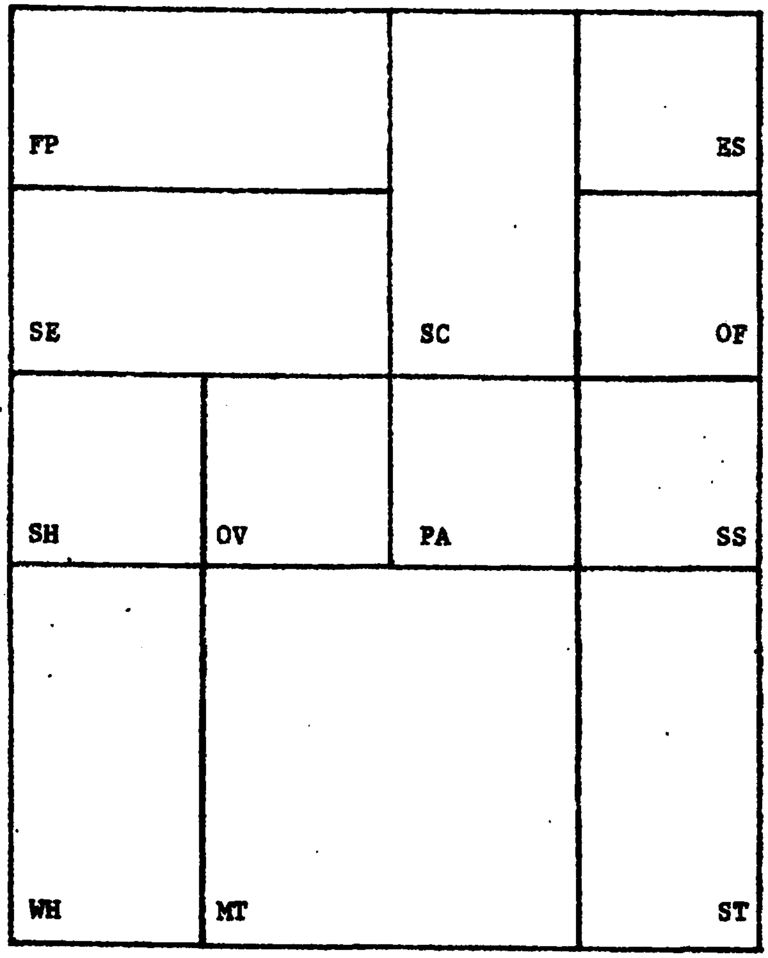


Figure 16. Slightly Improved Layout From Figure 15.

VI. SUMMARY AND CONCLUSION

The MATCH layout system is a prototype system developed to test the concept of interfacing optimization models with colorgraphic computers to interactively generate block layouts. The graph manipulation concept can be viewed as a generalization of the relationship diagram concept of Muther [3] combined with a sophisticated optimization model. The human decision maker still maintains the central role but has more power at his disposal to aid with the combinatorial aspects of the problem.

As with most aid to design, it is possible to evaluate the performance of the approach only in a fairly subjective fashion. We have the ability to call the CRAFT program to attempt an improvement of any layout generated using MATCH. We score the layouts using both adjacency and distance measures. While MATCH was designed primarily to model adjacency relationships, it also seems to perform very well with respect to distance based scores. It is seldom possible to improve significantly on any of the scores using CRAFT. When the CRAFT interchanges do result in an improved score it is almost always at the expense of some badly contorted blocks.

We have generally been very pleased with the quality of layouts which we were able to generate with MATCH. While there are still significant improvements to be made in both the optimization modelling and the system implementation, this seems the most promising approach available for attacking the very difficult block layout problem.

References

1. Edmonds, J., "Paths, Trees, and Flowers," Canadian Journal of Mathematics, Vol. 17, 1965, pp. 449-467.

2. Foulds, L. R., "The Facilities Design Problem: A Survey," Management Science, to appear, 1982-1983.

3. Muther, Richard D., "Systematic Layout Planning," Industrial Education Institute, Boston, Massachusetts, U.S.A., 1961.

4. Tompkins, J. A. and J. M. Moore, "Computer Aided Layout: A User's Guide," AIIE, Norcross, Georgia, U.S.A., 1978.